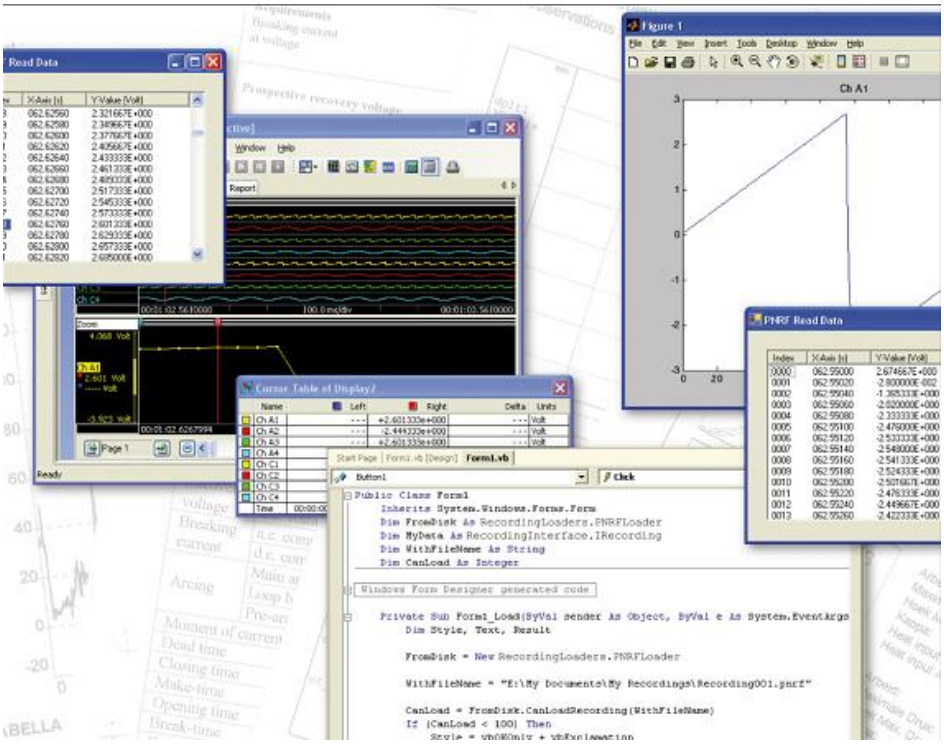
Genesis HIGH SPEED

Training

English



Perception PNRF Reader Asynchronous dataTraining

Document version 1.0 – November 2020

For Perception 7.18 or higher

For HBK's Terms and Conditions visit www.hbm.com/terms

Hottinger Brüel & Kjaer GmbH
Im Tiefen See 45
64293 Darmstadt
Germany
Tel: +49 6151 80 30
Fax: +49 6151 8039100
Email: info@hbm.com
www.hbm.com/highspeed

Copyright © 2020

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

LICENSE AGREEMENT AND WARRANTY

For more information about LICENSE AGREEMENT AND WARRANTY refer to:

www.hbm.com/terms

(Blank **Left** page only)

BLANK PAGE

Table of Contents

LAB 10 – Creating example of reading asynchronous data from a PNRF file

Start with *PnrfReader 2*

You can use the property ***TimeInfo*** of a data source (**IDataSrc**) to see the type of data it contains:

```
public interface IDataSrc
{
    …
    DataSourceTimeInfo TimeInfo
    …
}

There are three different time info's

public enum DataSourceTimeInfo
{
    DataSourceTimeInfo_Implicit = 0,
    DataSourceTimeInfo_Explicit = 1,
    DataSourceTimeInfo_External = 2
}
```

The first two enumerates are indicating that the data is synchronous, the last one (DataSourceTimeInfo_External) indicates asynchronous data. In this case each data sample has a Y-value and a time stamp.
The function WaveformWithTimes is used to get both the Y-values and time stamps. In the function below the Y-values are i

The following function is used to get the asynchronous data:

```
void WaveformWithTimes(DataSourceResultType ResultDataType, int FirstSample,
                int ResultCount, int Reduction, out object Data, out object Times);
```

This function will be used in the code below.

Purpose:

Creating pnrf reader program reading asynchronous data samples

- The code is now ready; you can compile it and use the debugger to examine your code.

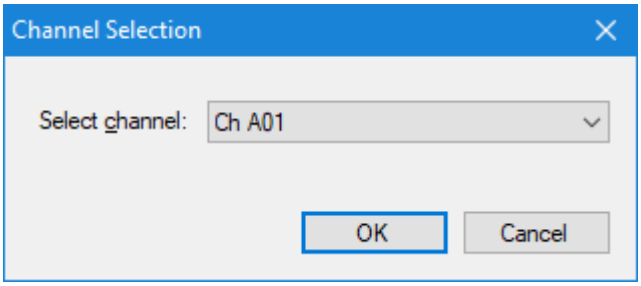
Result in *PnrfReader 10*

```
private void ShowRecordingData()
{
    PNRFLoader FromDisk = new PNRFLoader();

    string strRecordingFileName = edtSourceFileName.Text;
    IRecording myData = FromDisk.LoadRecording(strRecordingFileName);
    lblOutput.Text = myData.Title;
    listBox1.Items.Clear();

    if (myData.Channels.Count < 1)
    {
        MessageBox.Show("No Data");
        return;
    }
```

```
int nIndex = 1;
using (ChannelSelectionDialog dialog = new ChannelSelectionDialog())
{
    dialog.cbxChannelSelection.Items.Clear();
    for (int i = 1; i <= myData.Channels.Count; i++)
    {
        dialog.cbxChannelSelection.Items.Add(myData.Channels[i].Name);
    }
    dialog.cbxChannelSelection.SelectedIndex = 0;
    if (dialog.ShowDialog(this) == DialogResult.OK)
    {
        nIndex = dialog.cbxChannelSelection.SelectedIndex + 1;
    }
    else
    {
        return;
    }
}
IDataChannel myChannel = myData.Channels[nIndex];
listBox1.Items.Add(string.Format("Recording: {0}", myChannel.Recording.Title));
listBox1.Items.Add(string.Format("Recorder: {0} Channel: {1}",
                myChannel.Recorder.Name, myChannel.Name));
IDataSrc myDataSrc =
         myChannel.get_DataSource(DataSourceSelect.DataSourceSelect_Mixed);
bool isAsyncData = false;
if (myDataSrc.TimeInfo == DataSourceTimeInfo.DataSourceTimeInfo_Explicit)
{
    isAsyncData = true;
    listBox1.Items.Add(string.Format(
         "Data is asynchronous, data points are defined by Y value and explicit
          time stamp"));
}
else
{
    listBox1.Items.Add(string.Format(
                    "Data is synchronous, data points are equidistant"));
}

double dStartTime = myDataSrc.Sweeps.StartTime;
double dEndTime = myDataSrc.Sweeps.EndTime;

listBox1.Items.Add(string.Format("Recording Start: {0}, End: {1}",
              dStartTime, dEndTime));

listBox1.Items.Add("");
object mySegmentsData = null;
myDataSrc.Data(dStartTime, dEndTime, out mySegmentsData);

if (mySegmentsData == null)
{
    MessageBox.Show("No Data");
    return;
}

IDataSegments mySegments = mySegmentsData as IDataSegments;
if (mySegments == null)
{
    MessageBox.Show("No Segments");
    return;
}
if (mySegments.Count < 1)
{
    MessageBox.Show("No Segments");
    return;
}

IDataSegment mySegment = mySegments[1];

object oRawData = null;
object oTimes = null;

if (isAsyncData)
{
    mySegment.WaveformWithTimes(
          DataSourceResultType.DataSourceResultType_Double64, 1,
          mySegment.NumberOfSamples, 1, out oRawData, out oTimes);
}
else
```

```
        {
            mySegment.Waveform( DataSourceResultType.DataSourceResultType_Double64,
                 1, mySegment.NumberOfSamples, 1, out oRawData);
        }


        if (oRawData == null)
        {
            MessageBox.Show("No Raw Data");
            return;
        }

        double[] aSamples = oRawData as double[];
        double[] aX = null;
        if (isAsyncData)
        {
            aX = oTimes as double[];
        }
        double X0 = mySegment.StartTime;
        double DeltaX = mySegment.SampleInterval;
        double X, Y;

        for (int i = 0; i < aSamples.Length; i++)
        {
            if (isAsyncData)
            {
                X = aX[i];
            }
            else
            {
                X = X0 + i * DeltaX;
            }

            Y = aSamples[i];
            listBox1.Items.Add(string.Format("{0:000}:  X = {1}, Y = {2}", i, X, Y));

            // Do not show more then 500 samples
            if (i >= 500)
                break;
        }
    }
```

The Channel Selection dialog looks like:



In the combobox you can select a channel from the pnrf file.

Head Office
HBM
Im Tiefen See 45
64293 Darmstadt
Germany
Tel: +49 6151 8030
Email: info@hbm.com

France
HBM France SAS
46 rue du Champoreux
BP76
91542 Mennecy Cedex
Tél:+33 (0)1 69 90 63 70
Fax: +33 (0) 1 69 90 63 80
Email: info@fr.hbm.com

UK
HBM United Kingdom
1 Churchill Court, 58 Station Road
North Harrow, Middlesex, HA2 7SA
Tel: +44 (0) 208 515 6100
Email: info@uk.hbm.com

USA
HBM, Inc.
19 Bartlett Street
Marlborough, MA 01752, USA
Tel : +1 (800) 578-4260
Email: info@usa.hbm.com

PR China
HBM Sales Office
Room 2912, Jing Guang Centre
Beijing, China 100020
Tel: +86 10 6597 4006
Email: hbmchina@hbm.com.cn

© Hottinger Baldwin Messtechnik GmbH. All rights reserved.
All details describe our products in general form only.
They are not to be understood as express warranty and do
not constitute any liability whatsoever.

measure and predict with confidence